



Fundamentos de Big Data

Lección 3: Principales Tipos de Gráfico

ÍNDICE

Lección 3. – Principales Tipos de Gráficos	2
Presentación y objetivos	2
1. Principales Tipos de Gráficos [1]: Histogramas	3
2. Principales Tipos de Gráficos [2]: Scatter Plots.....	9
3. Principales Tipos de Gráficos [3]: Pie Chart	13
4. Principales Tipos de Gráficos [4]: Varios Plots.....	17
5. Principales Tipos de Gráficos [5]: 3-Dimensiones.....	19
6. Puntos clave	21

Lección 3. – Principales Tipos de Gráficos

PRESENTACIÓN Y OBJETIVOS

Esta asignatura contiene una gran cantidad de elementos teóricos, que han sido sintetizados para tratar de hacer cosas prácticas.

Antes de explicar algunos posibles Frameworks con los que se puede trabajar para hacer gráficos trataremos de explicar lo que es el Data Mining y el Análisis de Datos Exploratorio (EDA en inglés) con un Dataset típico al comenzar en Data Science, el Titanic Dataset.

En el caso que nos ocupa daremos una visión amplia de las posibles librerías que tenemos para hacer Gráficos en Data Science.



Objetivos

- Recordar conceptos básicos de gráficos con pandas, Matplotlib, Seaborn, etc. (Tomaremos como punto de partida en conocimientos el Iris Dataset ya visto)

1. PRINCIPALES TIPOS DE GRÁFICOS [1]: HISTOGRAMAS

Primeramente decir que cada tipo de proyecto tiene sus peculiaridades y según lo que necesitemos usaremos un tipo de gráfico u otro.

En este caso estaremos explicando algunas opciones que tenemos.

Y, nuevamente, con el Entorno de Jupyter Notebook.

PRINCIPALES DEPENDENCIAS

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
# para hacer gráficos interactivos
# %matplotlib notebook
# %matplotlib inline
```

Un posible dataframe (df)

```
In [2]: df = pd.DataFrame({"x": [1,32,4,23,40,2,2,27,6,18,49,67,46,7,
                                20,24,35,33,40,80,26,85,77,11,92,24],
                           "y": [31,10,85,25,4,83,32,43,66,18,93,6,42,
                                27,21,42,53,32,85,32,42,58,67,17,4,5]})
```

Figura 1.1: Tipos de Gráficos (parte 1)

Histogramas

Histograma 1

Casi sin parametrizar

```
In [3]: # histograma
plt.hist(df.y)
# cuadrícula
plt.grid(True)
# Eje X
plt.xlabel('Edad')
# frequency label
plt.ylabel('Frecuencia')
# plot title
plt.title('histograma 1')

# function to show the plot
plt.show()

# Nota: se refiere a frecuencia absoluta, repeticiones
```

Figura 1.2: Tipos de Gráficos (parte 2)

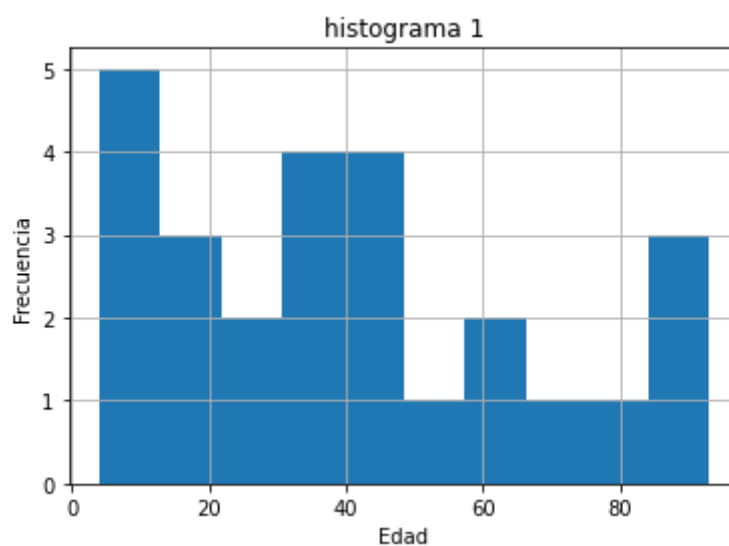


Figura 1.3: Tipos de Gráficos (parte 3)

Histograma 2

Con estas características:

- 10 divisiones
- 0.25 de ancho de la barra

```
In [4]: # plotting a histogram
plt.hist(df.y, bins=10,
         color = 'green',
         histtype = 'bar',
         rwidth = 0.25)

# cuadrícula
plt.grid(True)
# Eje X
plt.xlabel('Edad')
# frequency label
plt.ylabel('Frecuencia')
# plot title
plt.title('histograma 2')

# function to show the plot
plt.show()
```

Figura 1.4: Tipos de Gráficos (parte 4)

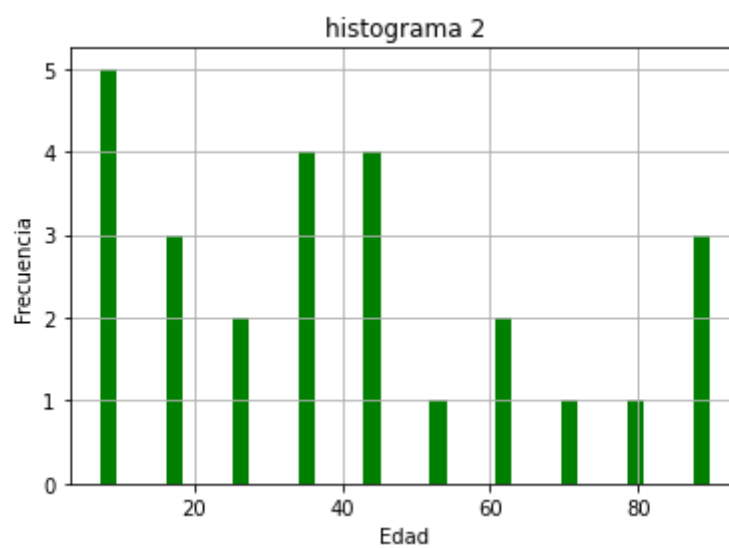


Figura 1.5: Tipos de Gráficos (parte 5)

Histograma 3

Con estas características:

- 10 divisiones
- 0.75 de ancho de barra

```
In [5]: # histograma
plt.hist(df.y, bins=10,
         color = 'green',
         histtype = 'bar',
         rwidth = 0.75)

# cuadrícula
plt.grid(True)
# Eje X
plt.xlabel('Edad')
# frequency label
plt.ylabel('Frecuencia')
# plot title
plt.title('histograma 3')

plt.show()
```

Figura 1.6: Tipos de Gráficos (parte 6)

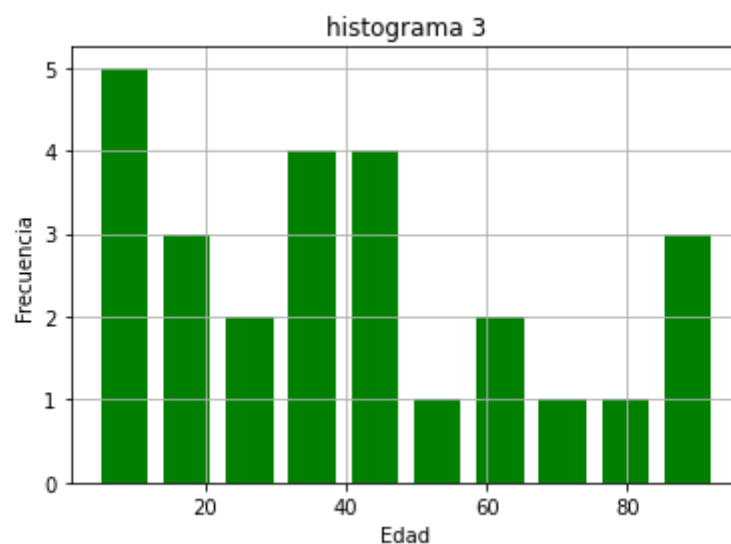


Figura 1.7: Tipos de Gráficos (parte 7)

Histograma 4

Con estas características:

- 3 divisiones
- 0.75 de ancho de barra

```
In [6]: plt.hist(df.y, bins=3,  
                color = 'green',  
                histtype = 'bar',  
                rwidth = 0.75)  
  
# cuadrícula  
plt.grid(True)  
# Eje X  
plt.xlabel('Edad')  
# frequency label  
plt.ylabel('Frecuencia')  
# plot title  
plt.title('histograma 4')  
  
# function to show the plot  
plt.show()
```

Figura 1.8: Tipos de Gráficos (parte 8)

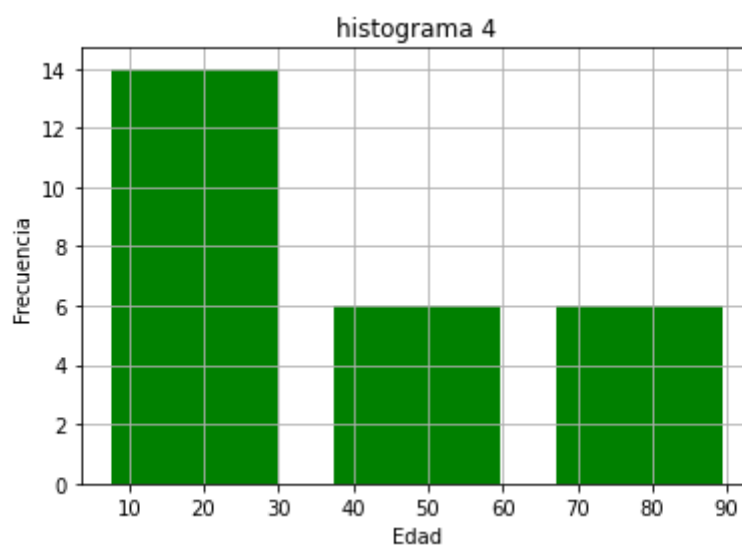


Figura 1.9: Tipos de Gráficos (parte 9)

del 10 al 30 hay 14????

```
In [7]: df = pd.DataFrame({"x": [1,32,4,23,40,2,2,27,6,18,49,67,46,7,
                                20,24,35,33,40,80,26,85,77,11,92,24],
                           "y": [31,10,85,25,4,83,32,43,66,18,93,6,42,
                                27,21,42,53,32,85,32,42,58,67,17,4,5]})
df.head()
```

```
Out[7]:
```

	x	y
0	1	31
1	32	10
2	4	85
3	23	25
4	40	4

```
In [8]: menores=0
for numero in df.x:
    if numero<=30:
        menores+=1

print(menores)
```

14

Sí !! ha cumplido con lo previsto

Figura 1.10: Tipos de Gráficos (parte 10)

2. PRINCIPALES TIPOS DE GRÁFICOS [2]: SCATTER PLOTS

Scatter plot

```
In [9]: # mismos datos
df = pd.DataFrame({"x": [1,32,4,23,40,2,2,27,6,18,49,67,46,7,
                        20,24,35,33,40,80,26,85,77,11,92,24],
                  "y": [31,10,85,25,4,83,32,43,66,18,93,6,42,
                        27,21,42,53,32,85,32,42,58,67,17,4,5]})
```

Scatter plot 1

```
In [10]: plt.scatter(df.x, df.y)
plt.show()
```

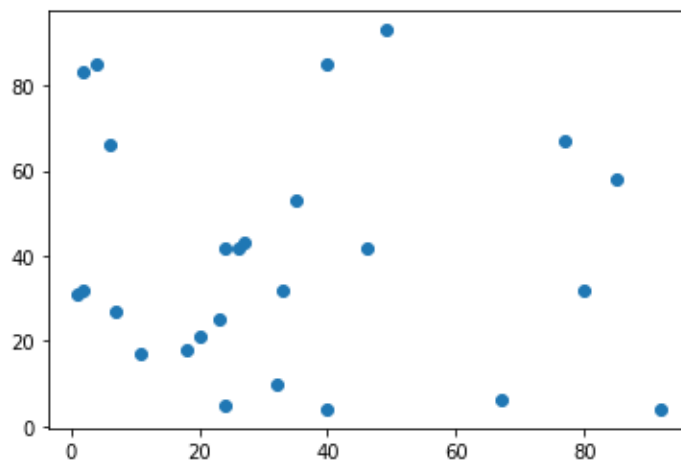


Figura 2.1: Tipos de Gráficos (parte 11)

scatter plot 2

```
In [11]: # Con otros datos !
```

```
In [12]: x = [1,2,3,4,5,6,7,8,9,10]  
y = [1,4,9,15,20,34,62,78,84,103]
```

```
In [13]: plt.scatter(x, y)  
plt.grid(True)  
plt.show()
```

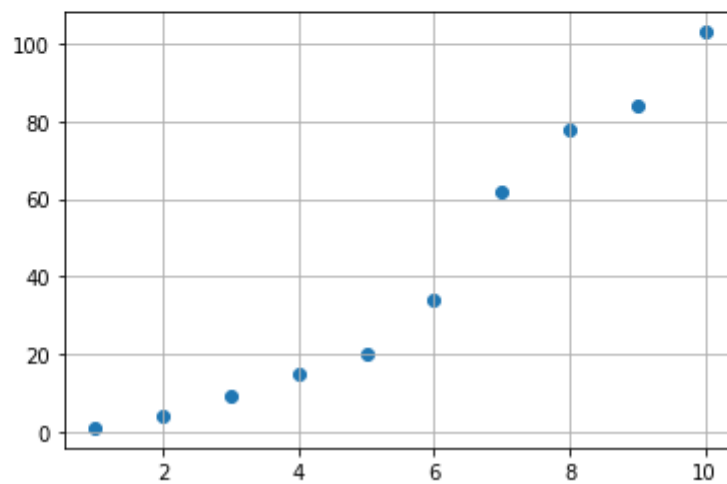


Figura 2.2: Tipos de Gráficos (parte 12)

Scatter plot 3

Con las siguientes características:

- label = "años"
- color = verde
- símbolo *
- s=500 es bastante grande

```
In [14]: plt.scatter(x, y,  
                    label= "años", color= "green",  
                    marker= "*", s=500)  
  
plt.xlabel('eje X')  
plt.ylabel('eje Y')  
  
plt.title('Mi scatter plot!')  
  
plt.legend()  
  
plt.show()
```

Figura 2.3: Tipos de Gráficos (parte 13)

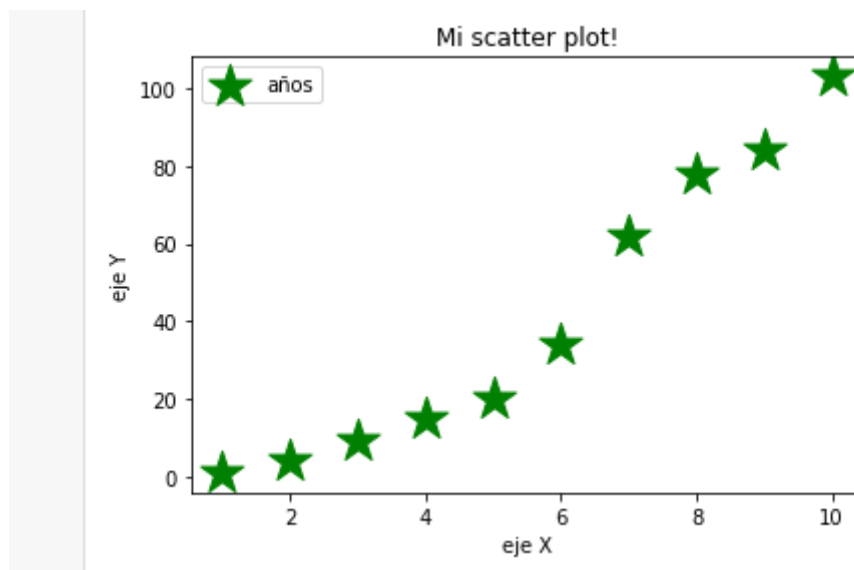


Figura 2.4: Tipos de Gráficos (parte 14)

Scatter plot 4

Con las siguientes características:

- s=100 es algo más pequeño

```
In [15]: plt.scatter(x, y,
                    label= "stars", color= "green",
                    marker= "*", s=100)

plt.xlabel('eje X')
plt.ylabel('eje Y')

plt.title('Mi scatter plot!')

plt.legend()

plt.show()
```

Figura 2.5: Tipos de Gráficos (parte 15)



Figura 2.6: Tipos de Gráficos (parte 16)

3. PRINCIPALES TIPOS DE GRÁFICOS [3]: PIE CHART

Pie chart (gráfico circular)

```
In [16]: y = ['fútbol', 'baloncesto', 'Gimnasio', 'TV']  
x = [4, 2, 4, 1]
```

Pie chart 1

Primera aproximación

```
In [17]: plt.title("Horas semanales de tiempo libre")  
plt.pie(x)  
plt.show()
```

Horas semanales de tiempo libre



Figura 3.1: Tipos de Gráficos (parte 17)

Pie chart 2

Algunas mejoras

```
In [18]: plt.title("Horas semanales de tiempo libre")  
plt.pie(x, labels = y)  
plt.show()
```



Figura 3.2: Tipos de Gráficos (parte 18)

Pie chart 3

Algo más complejo

Características:

- Se indican los colores
- le digo que empiece a plotear desde un ángulo de 90°
- radio del pie chart 1.0

```
In [19]: y = ['fútbol', 'baloncesto', 'Gimnasio', 'TV']  
x = [4, 2, 4, 1]  
  
plt.title("Horas semanales de tiempo libre")  
plt.pie(x, labels = y,  
        colors = ["r", "y", "g", "b"],  
        startangle = 90,  
        radius = 1.0)  
  
plt.show()
```

Figura 3.3: Tipos de Gráficos (parte 19)



Figura 3.4: Tipos de Gráficos (parte 20)

Pie chart 4

autopct le indica que escriba el número dentro (%)

```
In [20]: y = ['fútbol', 'baloncesto', 'Gimnasio', 'TV']
x = [4, 2, 4, 1]

plt.title("Horas semanales de tiempo libre")
plt.pie(x, labels = y,
        colors = ["r", "y", "g", "b"],
        startangle = 90,
        radius = 1.2,
        autopct="%1.2f%%")

plt.show()
```

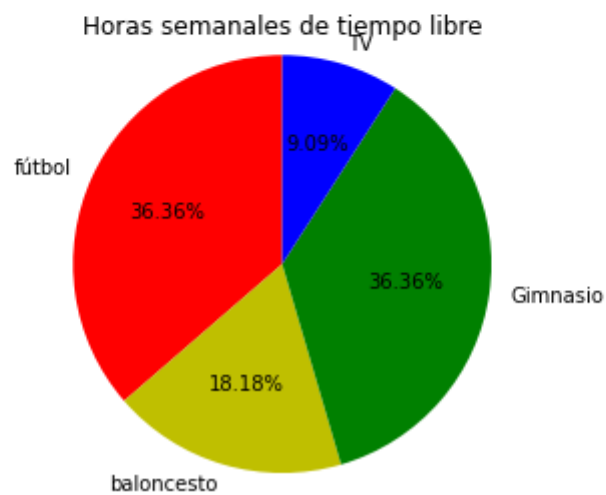


Figura 3.5: Tipos de Gráficos (parte 21)

4. PRINCIPALES TIPOS DE GRÁFICOS [4]: VARIOS PLOTS

Varios plots

```
In [21]: import numpy as np
plt.figure(figsize=(12,8))
# Lo metes todo en una figura para que no se te quede pequeño..

x = np.arange(-10, 10, 0.01)
# simplemente le das el rango (-10,10) de 0.01 en 0.01

# simplemente le dices los subplots
# (número de filas, número de columnas, número de grafico)
plt.subplot(2,2,1)
y1=x
plt.plot(x, y1)
plt.title("y=x")

plt.subplot(2,2,2)
y2=x**2
plt.plot(x, y2)
plt.title("y es x al cuadrado")

plt.subplot(2,2,3)
y3=x**3
plt.plot(x, y3)
plt.title("y es x al cubo")

plt.subplot(2,2,4)
y4=x**4
plt.plot(x, y4)
plt.title("y es x a la cuarta")

plt.show()
```

Figura 4.1: Tipos de Gráficos (parte 22)

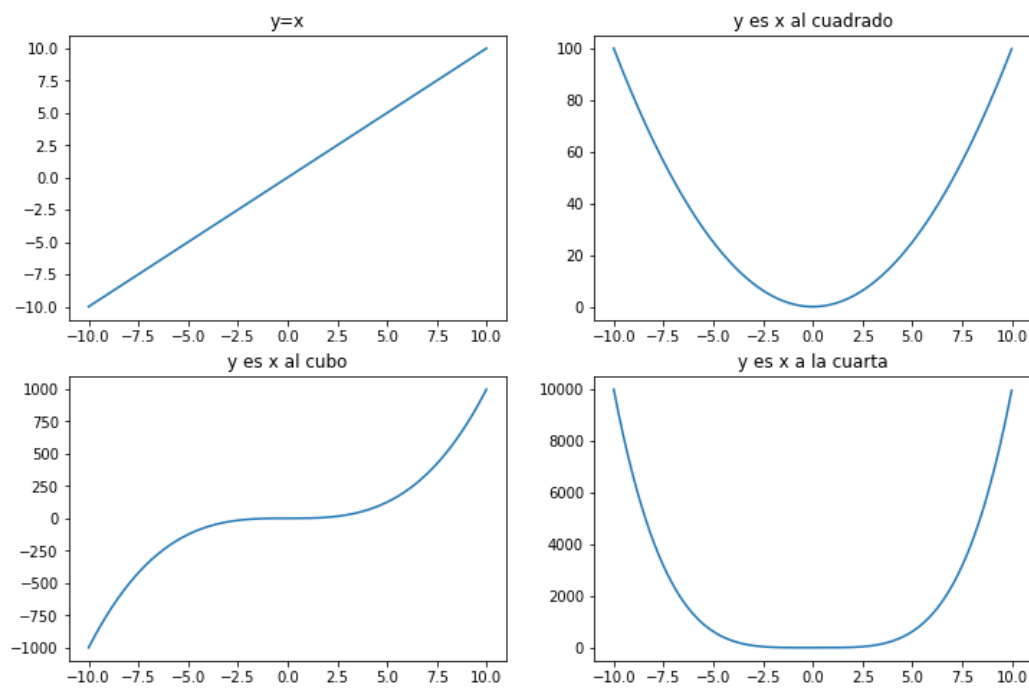


Figura 4.2: Tipos de Gráficos (parte 23)

5. PRINCIPALES TIPOS DE GRÁFICOS [5]: 3-DIMENSIONES

Gráficos en 3 Dimensiones

```
In [22]: # esto se necesita para usar el projection="3d"
from mpl_toolkits import mplot3d
ax = plt.axes(projection='3d')

# Data for three-dimensional scattered points
x =[1,2,3,4,5,6,7,8,9,10]
y =[4,4,4,2,4,5,2,3,5,6]
z =[3,3,3,3,5,5,2,4,7,8]
ax.scatter3D(x, y, z, marker="o", s=100)

plt.show()
```

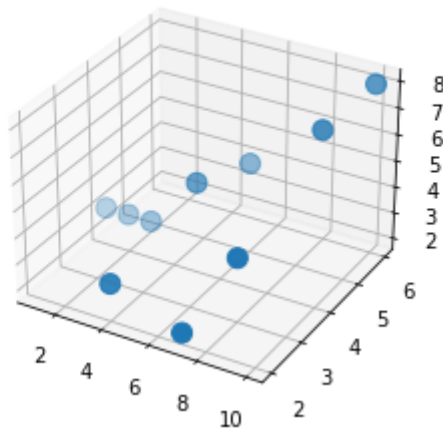


Figura 5.1: Tipos de Gráficos (parte 24)

o incluso algo más grande..

```
In [23]: plt.figure(figsize=(12,8))

# esto se necesita para usar el projection="3d"
from mpl_toolkits import mplot3d
ax = plt.axes(projection='3d')

# Data for three-dimensional scattered points
x = [1,2,3,4,5,6,7,8,9,10]
y = [4,4,4,2,4,5,2,3,5,6]
z = [3,3,3,3,5,5,2,4,7,8]
ax.scatter3D(x, y, z, marker="o", s=300)

plt.show()
```

Figura 5.2: Tipos de Gráficos (parte 25)

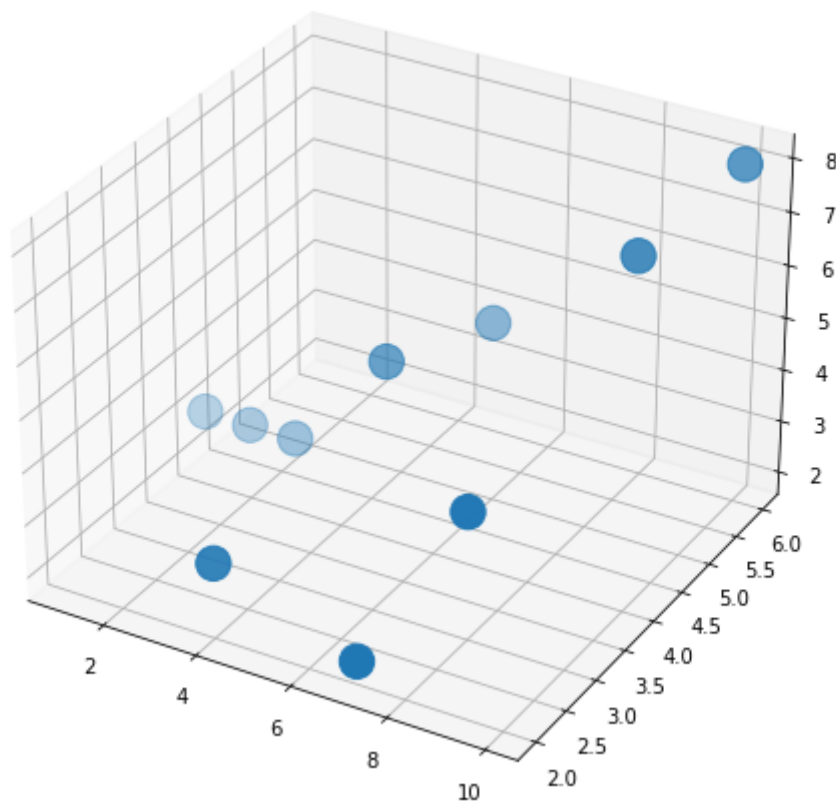


Figura 5.3: Tipos de Gráficos (parte 26)

6. PUNTOS CLAVE

- | Existen muchos tipos de Gráficos, teniendo cada uno sus peculiaridades.
- | Matplotlib pudiera ser suficiente para hacer cualquiera de ellos, aunque si queremos algo profesional podemos usar algún Framework.

